A New Navigation/Traveling Method in Virtual Environment

Tae-Wook Kwon† and Yoon-Chul Choy††

ABSTRACT

An important feature of virtual reality is the facility for the user to move around a virtual environment in a natural and easily controlled manner, Navigation. Navigation involves changing the perspective of the user in the virtual environment (VE). Natural locomotion methods are able to contribute to a sense of presence and reality. This paper focuses on the navigation method in the virtual environment, one of the major interfaces for the interactivity between human and virtual environments in virtual reality circumstances and worlds. It proposes a new navigation method: Intelligent Cruise-Control Navigation (ICCN), which provides a natural and user-centered navigation method in virtual environment and can improve the reality and the presence. Intelligent Cruise-Control Navigation is composed of three major phases: Constant Velocity Navigation, Collision Detection and Avoidance, and Path Adjustment. The ICCN can reduce the user's fatigue and improve the user's presence and reality in the virtual environment. Through the experimental study it has been determined that the ICCN will be a natural, straightforward, and useful interface in VE.

가상공간에서 새로운 이동기법에 관한 연구

권태욱† · 최윤철††

요 약

본 논문은 인간과 VR환경과의 상호작용을 지원하는 중요한 요소 중 하나인 이동의 문제점 및 해결 방법에 대하여 연구하였다. 이 논문에서 제시된 Intelligent Cruise-Control Navigation (ICCN)은 다중 사용자 환경의 가상공간을 이동할 때, 실세계와 유사한 이동방법을 제공하여 사용자로 하여금 가상공간에 대한 현장감 및 현실감의 제고에 초점을 두었다. ICCN은 사용자들이 가상공간에서 이동 시 부가적인 입력이 없이도 일정한 속도로 이동을 지원하는 Constant Velocity Navigation, 이동 중 장애물 및 다른 사용자(아바타)와의 충돌현상을 감지 및 회피하는 Collision Detection and Avoidance, 그리고 충돌회피 후 기존 방향으로의 계속된 이동을 지원하는 Path Adjustment 등의 기능을 제공한다. ICCN은 사용자의 부가적인 노력의 감소 및 병행작업을 보장, 현실과 유사한 사용자 중심의 navigation 기법을 제공, 가상공간과 현실과의 괴리를 줄임으로써 가상현실이 추구하는 현실감 및 현장감을 높일 수 있도록 하였다. 실험을 통하여 본 연구에서 제안한 ICCN이 사용자 중심의 매우 자연스럽고, 쉽고 편리한 가상공간 navigation 인터페이스라는 평가를 얻었다.

1. Introduction

The advancement of Virtual Reality, the World Wide Web (WWW), and Computer Graphics technologies have offered a new environment to us. VR provides computer-modeled environments where we can experience situations that feel like reality. For instance, VR can create circumstances or worlds that do exist (shopping, car driving, auto racing) or do not (space travel, traveling in dream

본 논문은 '97~'99년 공업기반기술개발사업의 지원에 의해 수행됨

† 준회원, 연세대학교 컴퓨터과학과 박사과정
†† 종신회원, 연세대학교 컴퓨터과학과 교수

like reality, traveling in the third world). VR can take us to places about which we can only imagine. We call these circumstances and worlds virtual environments (VE).

The final and conclusive goal of VR is to provide virtual environments or circumstances from which we can gain some feelings and experiences that are quite similar to the real things. We named these kinds of feelings and experiences as the Reality, the Presence and the Immersion. They are dependent on and influenced by many elements and conditions that are provided by the virtual environment, and much research has been conducted. In a part of the research, they have tried to improve the Reality and Presence by providing the user with useful and effective navigation interfaces such as walking, flying, riding, and driving modes.

An important feature of virtual reality is the facility for the user to move through a virtual environment in a natural and easily controlled manner, Navigation. Navigation, which is also called locomotion, travel or motion, involves changing the perspective of the user in the Virtual Environment. It allows the users to move in the Virtual Environment as well as reorient themselves to look at the world differently. Natural locomotion methods are able to contribute to a sense of presence and reality, this has been cited by some researchers as a defining attribute of VR.

The illusion of presence can be lost through unnatural experiences during travel in VE. This can be caused by poor interactive metaphors or by experiences, which do not agree with the user's everyday understanding of the real world. Several attempts have been made to develop new metaphors for walking through virtual environments. However, the intuitive metaphors described so far only can solve some of the problems. The other part concerns how to provide a virtual environment with more realistic properties so that the user's movement can be more natural and comfortable.

This paper focuses on the navigation method in the virtual environment, which is one of the major interfaces for the interactivity between human and virtual environments. It proposes a new navigation method: Intelligent Cruise-Control Navigation (ICCN), which provides a natural navigation method in virtual environments, closer to how we walk or move in the real world. It can then improve the Presence and Reality, the final goal of Virtual Reality. The ICCN is composed of three major phases. The first phase is Constant Velocity Navigation (CVN), which supports continuous and automatic constant travel and navigation services without any additional input from the user. The second phase is Collision Detection and Avoidance (CDA), which detects and avoids collision situations with a virtual object or any other user's avatar, by systematically supporting the user's avatar to bypass the object or the other user's avatar. The last phase is Path Adjustment; this supports the user's avatar to maintain it and return it to its original travel and navigation direction after the collision avoidance step.

2. Related Works

The most common task in VE's is that of navigating around the space of the environment. Some artificial methods must be provided for the user to move through the space, such as walking, flying, and physical user motion with treadmills, roller skates, or bicycles[8]. A number of researchers have addressed the issues related to navigation and travel in both the immersive virtual environments and in general 3D-computer interactions. They have insisted that studying and understanding human navigation and motion control is of great importance for comprehending how to build an effective virtual environment travel interface.

Wayfinding issues have been the subjects of studies by Darken and Sibert[1,2]. The use of maps, breadcrumbs, and landmarks were evaluated as tools for finding a path through a virtual environ-

ment. Their research shows that subjects in the treatment without any additional cues were often disoriented and found it extremely difficult to complete the task. For effective navigation, the results suggest that users of large-scale virtual worlds require structure, augmentations such as direction indicators, maps, and path restriction can all greatly improve both the wayfinding performance and the overall user satisfaction.

Xiao[3] presents a new technique for controlling a user's navigation in a virtual environment. It introduces artificial force fields, which act upon the user's virtual body such that the user is guided around obstacles, rather than penetrating or colliding with them. Li[9] describes an auto-navigation system, in which several efficient path-planning algorithms adapted from robotics are used. Additional techniques exist for general navigation within virtual environments. For example, terrain following is simple to perform for surfaces such as height fields over regular grids where there is a single point of intersection with the line extending vertically downwards from the viewpoint. In this case, the surface face to follow can be determined by the extent of the height field and the grid spacing.

Bowman[4,5] presents a categorization of techniques for first-person motion control, or travel, through immersive virtual environments, as well as a framework for evaluating the quality of different techniques for specific virtual environment tasks. Results indicate that pointing techniques are relatively more advantageous compared to gaze-directed steering techniques for a relative motion task, and to those motion techniques which instantly teleport users to new locations and are correlated with increased user disorientation.

Satalich[7] studied the navigation and wayfinding in virtual reality environments. The tasks assigned to the subjects for investigation of navigational awareness utilized the following three measures: orientation, route estimation and Euclidean estimations. The results indicated that having a map before entering the virtual environment can improved the performance, but not for exploring the virtual environment.

Various metaphors for viewpoint motion and control in 3D environments have also been proposed. The flying, eyeball-in-hand, and scene-in-hand metaphors for virtual camera control are identified [10,12,17]. As an extension of the scene-in-hand metaphor, Pausch et al.[16] make use of a World-in-Miniature representation as a device for navigation and locomotion in immersive virtual environments. Mine [11] offers an overview of motion specification interaction techniques. The overview also discusses the issues concerning the implementation of such specification in immersive virtual environments. Several user studies concerning immersive travel techniques have been reported in the literature, such as those comparing different travel modes and metaphors for specific virtual environment applications. Physical motion techniques have also been studied, including an evaluation of the effect of a physical walking technique on the sense of presence[14,15].

3. Intelligent Cruise-Control Navigation (ICCN)

In the existing typical Desktop VR systems, if the user wants continuous travel or navigation then the user has to input each moving event continuously with a mouse or keyboard. The speed of travel or navigation in VE depends on the mouse-dragging distance starting at the first pressed point. In addition, to change from the walking-mode to the running-mode, the shift key needs to be pressed down also. When the avatar runs against any objects in the virtual environment the system generates a collision detection event and then the avatar is stalled and can not move any further. If the avatar meets with any other avatar in the multi-user virtual environment then the two avatars go through each other (see Fig. 1), because

Fig. 1. Collision with object and avatar in Active world.

the existing VR systems do not support the collision detection between the two avatars. These situations are very different from our real-life situations, and cause reduction of the Presence, Reality, and Immersion feelings for the VR users. The ICCN can solve such problems in the existing VR service systems and improve Reality and Presence for users, supporting navigation method similar to the real world.

The ICCN is named after the cruise control technique of a vehicle, it provides a continuous and constant driving speed service to the driver until the driver steps on the accelerator or break pedal. This driving service is a very comfortable and useful, for it can reduce a driver's fatigue and effort when driving a very long, straight and spacious road, such as an express way. The Intelligent Cruise-Control Navigation is composed of three major phases. The first phase is Constant Velocity Navigation (CVN), which supports continuous and automatic constant travel and a navigation service without any additional input from the user. The second phase is Collision Detection and Avoidance (CDA), it provides two steps: Collision Detection and Collision Avoidance. The first step detects and recognizes a collision situation with the virtual objects or the other avatars in the virtual environment. The second step avoids collision situation with a virtual object or any other user's avatar, by systematically supporting the user's avatar by-pass the object or the other user's avatar without stalling and passing through it. The last phase is Path Adjustment; this supports the user's avatar to maintain it and return it to its original travel and navigation direction after the previous collision avoidance step.

3.1 Constant Velocity Navigation (CVN)

In most of the existing VR service systems, the user controls the speed of navigation and travel with the mouse, keyboard or other devices (bike pedal, glove, joy stick, VMC). But if the user selects the ICCN mode, then it provides an automatic Constant Velocity Navigation function to the user. The CVN phase supports this constant and continuous travel as if the user was continuously giving moving events with the various input devices. If the CVN is serviced, it gives hands-free navigation. Hence, the user can handle other tasks as well, such as chatting, reading, or talking on the telephone, while traveling and navigating in the virtual environment, without inputting the moves in the VE. This ICCN mode operates in toggle mode, so if the user wants to turn off the ICCN mode then the user simply needs to select the ICCN button again. This CVN function can be used in various situations. For example, it can be used when the user wants to travel or walk in a very large and wide VE area or when the user wants to take a walk in the VE alone or with another user's avatar. It will also be very helpful to the novice or handicapped users, for they may have some difficulties in operating the mouse or keyboard.

In order to support CVN, we have to continuously change the user's location using the position variable in the viewpoint node in VRML. We divide the user's orientation (360^{0}) into 4 sections, and then continuously add to or subtract from the position value in the viewpoint node, depending upon the user's orientation. For instance, if the

user's orientation is $-$ 0.39249($-$ 22.5 ^{0}) then it conveys the plus sin(22.5) value to the X-axis value; the minus cos(22.5) value to the Z-axis value; in the position variable in the viewpoint nodé. If the user's orientation is +0.39249(+22.5 ^{0}) then it conveys the minus sin(22.5) value to the X-axis value; the minus cos(22.5) value to the Z-axis value; in the position variable in the viewpoint node. If the user's orientation is 1.96245(+112.5 ^{0}) then it conveys the minus sin(112.5) value to the X-axis value; the plus cos(112.5) value to the Z-axis value; in the position variable in the viewpoint node. If the user's orientation is 3.53241 (+202.5 ^{0}) then it simultaneously conveys the plus sin(202.5) value to the X-axis and plus cos(202.5) value to the Z-axis value in the position variable in the viewpoint node. While supported by the CVN service, the user can also change his navigation direction, by giving a direction change event with the typical input devices, like the user does in the typical virtual reality system.

3.2 Collision Detection and Avoidance (CDA)

In the Collision Detection and Avoidance phase, it first catches and estimates a collision situation with the virtual objects or the other users' avatars in the virtual environment. If the user's avatar runs against a virtual object then it generates a collision detection event, therefore CDA has to systematically catch or pre-estimate this collision detecting situation, using the virtual object's location data. It also generates a collision detection event when the user's avatar approaches the other user's avatar. To detect collision situation between two avatars, CDA uses the avatar's data information in the system, such as their ID, location, scale data, and so on. If two different avatars approach each other less than 1~2m apart then this phase recognizes that these two avatars will have a collision, and generates a collision detection event for the next step. After the Collision Detection, this phase starts the next Collision Avoidance step. This step assumes that the user's avatar will by-pass objects and the other user's avatar systematically and naturally until the avatar is moved to a distant location, not colliding with the object or the other users' avatar (see Fig. 2, 3).

The direction of any bypass generally depends upon how the user moves the user's avatar. However, CDA is used for the avatar to move automatically to go to the right or to the left according to the object's relative location to the user's navigation direction. If the object is on the right side of the user's navigation direction, the bypass will move to the left, and vice versa. At this step, when CDA receives or recognizes the collision detection event, it generates the right- or left- and forward- movement events making the avoiding paths look like an ellipse, until the collision

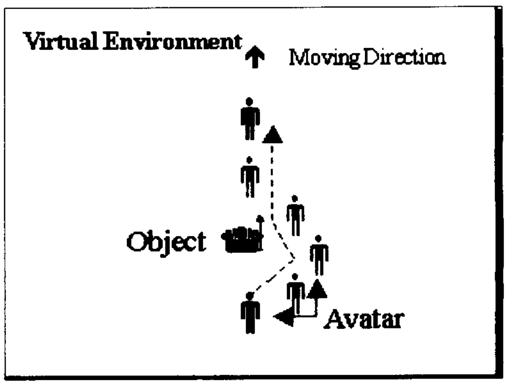


Fig. 2. CDA and Path Adjustment with object.

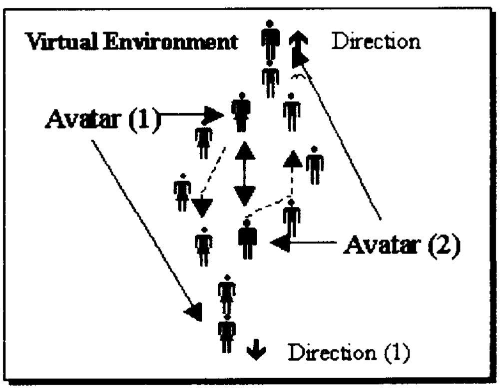


Fig. 3. CDA and Path Adjustment with Avatar.

detection event does not occur any more. The ellipse route is much more similar to a real life situation, and the avatar's behavior is smoother. At this phase the system counts the number of generated right or left and forward movement events for the next phase.

The CDA phase, the most time- and performance- consuming process in this research, uses the location and scale data of the other virtual avatars and the objects in the VE. It predefines the angle (azimuth) between the user and the other avatar or the virtual objects based on the user's orientation (Direction of movement) (see Fig. 4). The angle between the user and the object A (θ_a)/ object B (θ_b) is calculated by the JavaScript Math. atan function with the location values of the user (X_o, Z_o) and the object A (X_a, Z_a) / object B (X_b, Z_b). The angle of object A is $\tan^{-1}((X_a\ X_o)/(Z_a\ Z_o))$.

This collision detection process considers only the virtual objects, located in the focus area: less than 20 meters from the user's location. If a virtual object is in the focus area and the sin(θ) value is smaller than Repulsive Force Field (RFF)(1.5 times the object scale value) then the user's avatar and the object have a collision situation. In Fig. 4, the object A is in collision situation because the sin(θ_a) value is smaller than the radius of the circle A (RFF-a). However, the object B is not in collision situation, for the sin(θ_b) value is bigger than the RFF-b.

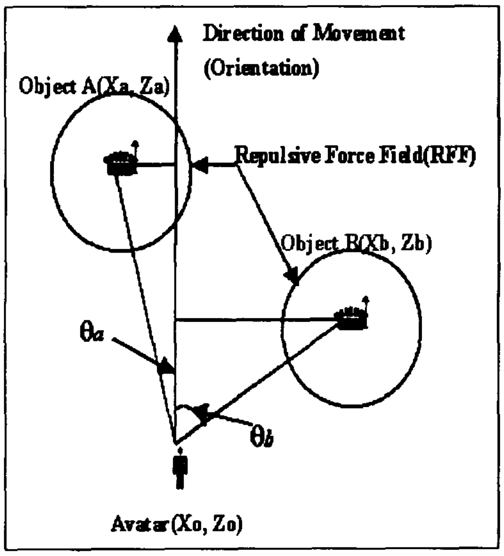


Fig. 4. Collision Detection with objects.

If the collision detection occurs, the phase starts the next collision-avoidance step. The goal of this step is to make this process appear more natural and much similar to the real life situation where we come across an object or other people on the street. Moreover, we try to make this avoidance path look like an ellipse based on the scale value of the virtual object and the avatar. The ellipse route is much more similar to a real life situation so it makes the avatar's behavior smoother and more natural. The avoidance step starts from the position, 1.5 meters plus the object's RFF value. This avoidance path will bypass the position 1.0 meters away from the object.

The direction of avoidance is dependent on both the sin() value and the user's orientation value. When the user's orientation value is in between +0.0 (+0°) and + 4.70988 (+270°) and the sin() value is smaller than the user's orientation value then we can estimate the object is on the right side of the user's locomotion way. Therefore, the left way collision avoidance is performed for this case. But when the user's orientation value is in between -0.0 (-0°) and 1.56996 (-90°) and the sin(θ) value is smaller than the user's orientation value, we can estimate the object is on the left side of the user's locomotion way. Thus, the right way collision avoidance is performed in this case. At this step, it generates the right- or left- and forward- movements, which makes the avoidance path look like an ellipse. The system also counts the number of generations for right- or left- and forward- movements for the next phase (Path Adjustment).

When the user's avatar meets with another avatar other than the virtual objects, then the process is the same as above, but is much simpler. When two avatars approach less than 2 meters from each other then it recognizes a collision

situation and they simply bypass by moving 1 meter to the right or left from the user's direction of locomotion depending on the other avatar's location.

3.3 Path Adjustment

The goal of this final phase is to prevent disorientation and spatial loss in the virtual environment, by the guarantee that the avatar keeps its original travel/navigation direction and path. This process depends on the original path of the user's avatar. That is, it is an original path-oriented process. In the previous phase, the system keeps the avatar's original position data and counts the number of generated right- or left- and forward- movement inputs during the Collision Avoidance process. If the CDA phase did not receive any more collision detection events then it starts the Path Adjustment process. This process uses the counted number of the right- or left- and forward- movements, which were saved at the second step in the previous phase. At the process, it generates the left- or right- and the forward-moving events to make the adjusting path follow an ellipse (see Fig. 2, 3). And the left- or right-moving events offset the generated counts of the right- or left- moving events in the previous Collision Avoidance step.

4. Results and Discussion

We developed a template Multi-user 3D VR system with JAVA and the commercial Cosmo Player browser. We created a virtual world; it has the boundary of 500 * 500 and the building of 160 * 60 with a cross shape corridor in the center. Moreover, it is composed of 250 virtual objects including 12 objects inside a building (see Fig. 5). The system provides both of the navigation methods; a typical navigation method based on the use of a mouse and keyboard and the ICCN navigation method, which has been proposed in this paper. Therefore the participants can also travel with the simultaneous use of typical navigation and ICCN navigation.

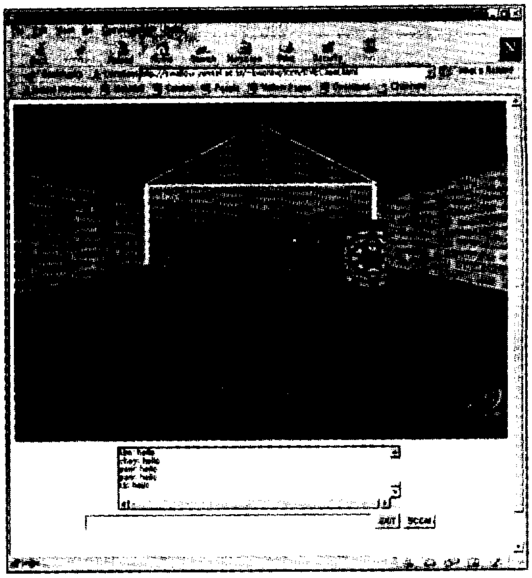


Fig. 5. ICCN Multi-user 3D VE system.

Twenty-one people took part in the experimental study; they are came from four different laboratories within the Computer Science Dept. of Yonsei University. All of the participants are graduate school students: 14 in Master's programs and 7 in their Doctoral programs. In addition, eight participants (5 from Master's programs, 3 from Doctoral programs) are researching in the same field, Virtual Reality.

In the post-experiment questionnaire five questions were asked in total. Three questions cover the following three aspects of navigation: general movement - how simple or complicated it was to move around; placement the difficulty in getting from one place to another; and how natural the movement was. The last two questions cover the effects of the research and the application area for the navigational interface. The questions give extra relative scores to the ICCN navigation method compared with the typical navigation method, which uses a mouse and keyboard. The questions and results are summarized in Table 1.

Table 1. Questions and Results of experiment.

General Navigation	Getting from Place to Place
Did you find it relatively "simple" or relatively "complicated" to move through the computer-generated world?	How difficult or straight-forward was it for you to get from a place to place?
To move trough the worlds was···	*To get from a place to place was···*
1.very complicated	1.very difficult
···	···
10.very simple	10.very straightforward
Result(mean, S.D)	**Result(mean, S.D)**
Mean: 8.075	Mean: 8.2
S.D: 1.172884	S.D: 0.978721
Natural/Unnatural	**Effects of Interface**
The act of moving from a place to place in the computer-generated world can seem to be relatively "natural" or "unnatural"?	Do you think this navigation interface is really helpful for the VE navigation?
The act of moving from a place to place seemed to me to be performed···	*For the navigation in VE, this interface is ···*
1.very unnatural	1.very not useful
···	···
10.very natural	10.very useful
Result(mean, S.D)	**Result(mean, S.D)**
Mean: 8.775	Mean: 8.475
S.D: 1.18627	S.D: 0.880714

For Question 1., the attendees gave scores in between 6.0 and 10.0 (mean: 8.075, S.D: 1.173). The reason they gave for the scores was that it does not require additional input for navigation. For Question 2., the attendees gave scores in between 6.0 and 10.0 (mean: 8.2, S.D: 0.979) also. The reason they put for the scores was that it also does not require additional input and it simultaneously supports both of the typical mouse- and keyboard-based navigation interface and this research's ICCN navigation interface. From the results of Question 1 and Question 2, we recognized that the navigation treatment in VE is not easy even for users familiar with computer environments. For Question 3.: natural or unnatural, the results showed that attendees have rated the interface from 7.0 to 10.0 (mean: 8.775, S.D: 1.186); it was a much higher result than we had expected. The major reason they put for the scores was that it supports automatic collision detection and avoidance with virtual objects and avatars. Overall, most of the attendees voted positively for the interface to be an effective and convenient means of Navigation in VE's. The reasons they presented for the results are the user-friendliness (easy to use), supporting of parallel tasks, and so on of the interface.

5. Conclusion and Future Works

The Intelligent Cruise-Control Navigation, which is composed of those three major phases mentioned above, implements the navigation metaphor of providing more user-centered and natural navigation methods in interactive virtual environments. It also gives more useful and effective navigation services to the user, solving unrealistic navigation situations, like when an avatar is stalled after a collision with virtual objects or when avatars go through each other when the avatar meets with other avatars. Those situations are happening in the present virtual reality service systems. ICCN can improve the user's virtual Presence and Reality, which is one of the ultimate goal of virtual reality.

One of the effects we expect from the research will be a reduction in the user's fatigue and a support of parallel tasks. In most of the existing VR systems, the user has to input the moving event with the various input devices continuously when the user wants to travel in the VE. Thus, for a 3D VR user to chat with other users, the user has to either stop and type in the conversations using both hands, or type with one hand and enter

each moving-event with a mouse or keyboard using the other hand. These situations are very unrealistic and difficult to manage. However, the ICCN will remove these difficult and unrealistic situations by providing an automatic constant and continuous velocity navigation service: Hands-free Navigation. The user can travel and chat with others in the VE with optimal convenience, and little effort.

The second effect we expect to gain from the research is an improvement in Presence and Reality in the VR. Navigation is one of the major interfaces in the VR; it can improve the Presence and Reality, by reducing the differences between the computer-modeled VE and the real environment, and by supporting a natural navigation method in VE. Because the ICCN tries to provide a user-centered and much similar navigation method to real life and eliminates the unrealistic conditions of some existing VR service systems, the ICCN will be an appropriate and convenient service tool. Therefore the ICCN can contribute to improvement of the user's Presence and Reality in the VR, by changing the navigation method and providing a somewhat lifelike travel and navigation experience in the VR for users.

From the experimental study, we gained very encouraging results for the general navigation, the natural navigation, and the navigation interface for the user in VE. For our future research, we will work towards developing a more natural collision-detection technique and a more natural and smoother S.P (system processor) in avoiding or bypassing virtual objects and avatars. Moreover, we will be concerned with effective interfaces between virtual avatars that can improve the virtual Presence and Reality of VEs as well.

References

[1] Darken, R. and Sibert, J. A Toolset for Navigation in Virtual Environment, Proceedings of ACM User Interface Software & Technology, pp. 157-165, 1993.

[2] Darken, R. and Sibert, J. Navigating in Large Virtual Worlds. The International Journal of Human-Computer Interaction, 8(1), pp. 49-72, 1996.

[3] Dongbo Xiao and Roger Hubbold. Navigation Guided by Artificial Force Fields, CHI'98, pp. 18-23, 1998.

[4] Doug A. Bowman and Larry F. Hodges. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environment. Proceeding of the symposium on Interactive 3D Graphics, pp. 35-38, 1997.

[5] Doug A. Bowman, David Koller, & Larry F. Hodges. Methodology for the Evaluation of Travel Technology for Immersive Virtual Environments. Virtual Reality: Research, Development, and Applications, vol. 3(2), pp. 120-131, 1998.

[6] Frederick P. Brooks Jr. Walkthrough Project: Final Technical Report to NSFC/ISE. Technical Report TR92-026, Computer Science Dept. UNC at Chapel Hill, 1992.

[7] Glenna A. Satalich. Navigation and Wayfinding in VR: Finding Proper Tools and Cues to Enhance Navigation Awareness. Master's thesis, University of Washington, 1995.

[8] H. Iwata and K. Matsuda. Haptic Walkthrough Simulator: Its design and application to studies on cognitive map. In The 2nd International Conference on Artificial Reality and Tele-existence, ICAT92, pp. 185-192, 1992.

[9] Tsai-Yen Li, Jyh-Ming Lien, Shih-Yen Chiu, and Tzong-Hann Yu. Automatically Generating Virtual Guided Tours. Computer Animation, 1999, Proceedings, pp. 99-106, 1999.

[10] J. Mackinlay, S. Card, and G. Robertson. Rapid Controlled Movement Through a Virtual 3D Workspace. *Proceedings of SIGGRAPH* (Dallas, TX, 1990), in *Computer Graphics*, vol. 24,

no. 4, pp. 171-176, 1990.

[11] M. Mine. Virtual Environment Interaction Techniques. UNC Chapel Hill Computer Science Technical Report. TR95-018, 1995.

[12] R. Pausch, T. Burnette, D. Brockway, and M. Weiblen Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures. *Proceedings of SIGGRAPH* (Los Angeles, CA), pp. 399-400, 1995.

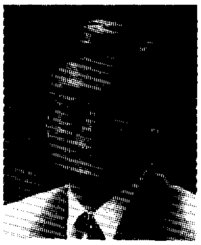
[13] W. Robinett and R. Holloway. Implementation of Flying, Scaling, and Grabbing in Virtual Worlds. *Proceedings of Symposium on Interactive 3D Graphics* (Cambridge, MA), pp. 189-192, 1992.

[14] Mel Slater and Martin Usoh. Representations systems, perceptual position, and presence in immersive virtual environments. Presence, 2(3), pp. 221-233, 1993.

[15] Mel Slater, Martin Usoh, and Anthony Steed. Taking steps: The influence of walking technique on presence in virtual reality. ACM Trans. On Computer-human Interaction, 2(3), pp. 201-219, 1995.

[16] Stoakley, R., Conway, M., Pausch, R. Virtual Reality on a WIM: Interactive Worlds in Miniature. In proc. CHI'95, pp. 265-272, 1995.

[17] E. Strommen. Children's Use of Mouse-Based Interfaces to Control Virtual Travel. *Proceedings of CHI* (Boston, MA), pp. 405-410, 1994.



최 윤 철

1773년 서울대학교 전자공학과 졸업(공학사)

1975년 6월 공학 석사 (Univ. of Pittsburgh)

1979년 6월 공학 박사 (Univ. of California, Berkeley, Dept. of IE&OR)

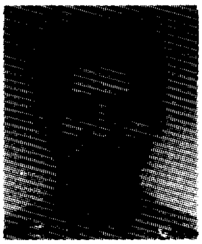
1979년 8월~1982년 7월 Lockheed사 및 Rockwell International사 책임 연구원

1982년 9월~1984년 1월 Univ. of Washington 전산학

1990년 9월~1992년 1월 Univ. of Massachusetts 연구교수

1984년~현재 연세대학교 컴퓨터과학과 교수

관심분야 : 멀티미디어, 하이퍼미디어, 지리정보 시스템 (GIS)



권 태 욱

1986년 3월 육군사관학교 졸업 (이 학사)

1995년 9월 미 해군대학원 컴퓨터공학 석사

1998년~현재 연세대학교 컴퓨터과학과 박사과정

관심분야 : 멀티미디어, 가상현실(VR), HCI, W.C (wearable computer).